

ARTÍCULOS DE TECNOLOGÍA &gt; FRONT END

# Creando y publicando una biblioteca Javascript en el NPM



Felipe Nascimento

25/10/2021



Pensando en una filosofía de código abierto donde todos pueden hacer uso de soluciones de manera gratuita y colaborativa, lo que vamos a hacer es aprender a crear una **biblioteca Javascript** y publicarla en el **NPM**, por donde todas las personas tendrán acceso.

En esta biblioteca vamos a tener una lista con los nombres de varios lenguajes de programación que podrán ser usados como base de datos. Para desarrollar ese proyecto vamos a necesitar del [node.js](#) y del [git](#) instalados en nuestra maquina.

## Clonando el repositorio

Para comenzar, creamos un repositorio en el **github** y marcamos la opción de inicializar con el README. Con eso podemos traer el repositorio para nuestro computador a través del comando **git clone <url del repositorio>**

Ya tenemos todo el ambiente configurado y el repositorio en nuestro computador. Ahora nos falta crear una cuenta en el NPM.

## Creando una cuenta en el NPM

Para que sea posible publicar nuestro paquete necesitamos crear una cuenta en el [npmjs](#). Llenamos un formulario simple y estamos listos para continuar. Pero ¿que es, ese tal NPM?

Me gusta bastante la definición del sitio de [nodejs](#):

*npm, abreviación de Node Package Manager, son dos cosas: primero, es un repositorio online para la publicación de proyectos Node.js de código abierto; segundo, es un utilitario de línea de comando para interactuar con el repositorio mencionado, lo que ayuda en la instalación de paquetes, en la administración de versiones y en la administración de dependencias. Después de tener un paquete que deseamos instalar, podemos instalar con un único comando de línea:*

Llegó la hora de poner las manos en la masa.

## Iniciando nuestro paquete

Dentro de nuestro repositorio vamos a realizar algunos comandos:

- npm set init.author.email "email"
- npm set init.author.name "nombre"
- npm set init.license "MIT"

Para indicar al NPM nuestro nombre, email y que nuestro proyecto tiene la [licencia MIT](#).

Luego después de esta etapa, tenemos que agregar nuestro usuario indicando que ese es nuestro paquete, a través del comando:

- npm adduser

El NPM va a hacer algunas preguntas como el nombre de usuario, la clave y el email registrado

El próximo paso es crear el **package.json**, es ese archivo que va contener todas las informaciones de nuestro paquete, como la versión, nombre, url del repositorio, entre otras. Nuevamente vamos a usar un comando de npm que nos va auxiliar en este proceso:

- **npm init**

El va a hacer una serie de preguntas:

- Nombre - Nombre del paquete
- Version - Version del paquete
- Descripcion - De que se trata el paquete
- Entry point - donde será ejecutada la primera instrucción. **Aquí vamos dejar como src/index.js, mas adelante vamos a crear esa carpeta y ese archivo.**
- Test command - Tiene relación con las pruebas, vamos a deja vacío por ahora
- Github repository - Repositorio del github
- Keywords - Palabras llaves van a facilitar a la hora de encontrar nuestra biblioteca
- License - licencia de la biblioteca

Ahora, si entramos en la carpeta de nuestro proyecto, vamos a ver que el archivo **package.json** está creado y si abrimos él con nuestro editor de texto (**VScode**) podemos ver que todas las configuraciones están allí.

Al fin, vamos a comenzar a crear nuestra biblioteca.

## Creando la biblioteca

Dentro de la carpeta de nuestro proyecto, vamos a crear otra carpeta llamada **src** y allá adentro creamos dos archivos:

**lenguajes-de-programacion.json** un **JSON** que posee el nombre de los lenguajes:

```
[ "JavaScript",  
  "Java",  
  "Python",  
  "C#",  
  "PHP",  
  "C++",  
  "C",  
  "Typescript",
```

```
"Ruby",  
"Swift",  
"Assembly",  
"Go",  
"Cobol",  
"R",  
"Pearl"  
];
```

Y nuestro archivo principal llamado **index.js**( recuerda del Entry Point? **src/index.js**). Aquí necesitamos hacer uso de **require** una función embutida del nodejs responsable por incluir módulos en archivos separados.

En nuestro caso ella está incluyendo **lenguajes-de-programacion.json** dentro del **index.js** para que podamos usarlo.

```
const lenguajes = require('./lenguajes-de-programacion.json')
```

Cuando las personas quieran utilizar nuestra biblioteca en sus proyectos, ellas necesitaran usar el **require** para incluir dentro del código, entonces, necesitamos exportar nuestro código para que el pueda ser importado en otro lugar, y para eso vamos a usar el [module.exports](#).

Dentro del `module.exports` tenemos un objeto que contiene una propiedad llamada **all** que va mostrar todas los lenguajes cuando sea ejecutada.

```
module.exports = {  
  all: lenguajes  
}
```

Nuestro código final quedó así:

```
const linguagens = require('./lenguajes-de-programacion.json')  
module.exports = {  
  all: lenguajes  
}
```

Si hacemos una prueba rápida en la consola del node haciendo un **require** en nuestro entry point y después ejecutando **.all**

Podemos ver que al llamar el **teste.all** conseguimos exhibir todas los lenguajes. Estamos casi al final.

Sin embargo, nuestro código está apenas en nuestro computador, tenemos que enviar esos archivos de vuelta al github.

## Agregando los archivos en el Github

Primero usamos el comando **git status** que nos va mostrar los archivos no monitorados, aquellos que creamos después de haber clonado el repositorio.

Necesitamos agregar a nuestro repositorio la carpeta **src/** y el archivo **package.json**. Podemos agregar de una solo vez con el comando **git add \*** y luego después hacemos un commit **git commit -m "comentário"** para salvar las modificaciones y finalmente, con el comando **git push**, mandamos las actualizaciones de vuelta para el repositorio del github.

Ahora falta el último paso, publicar la biblioteca en el NPM para que todas as personas puedan tener acceso.

## Publicando en el NPM

En la carpeta de nuestro proyecto basta digitar el comando **npm publish** y listo:

Si regresamos al sitio de npm podemos ver que nuestra biblioteca fue publicada con éxito.

Podemos salir de la carpeta de nuestro proyecto e instalar la biblioteca ejecutando el siguiente comando:

```
npm i nombre-de-los-lenguajes-de-programacion
```

Vamos a hacer una prueba

## Probando nuestra biblioteca

Después de instalar la biblioteca, creamos un nuevo archivo **index.js** y dentro de él hacemos un **require** de nuestro paquete e imprimimos a través del console.log:

```
const lang = require('nombre-de-los-lenguajes-de-programacion')
console.log( lang.all)
```

¡Uf! después de crear la cuenta en el github, en el npm, configurar el git y crear nuestro código Javascript, ¡conseguimos publicar nuestra primera biblioteca en el NPM!

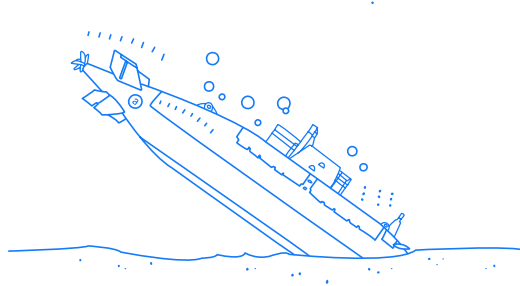
Y si quieres profundizarte mas en el front-end, aquí en **Alura Latam** tenemos una [formacion front-end](#) para que puedas comenzar de cero y convertirte en un profesional de punta.

Puedes leer también:

- [Convertir int a String en Java](#)

- 

¿Cuál curso estás buscando?

[INICIAR SESIÓN](#)[NUESTROS PLANES](#)[TODOS LOS  
CURSOS](#)[FORMACIONES](#)[CURSOS](#)[PARA EMPRESAS](#)

## ¡UPS! Ha ocurrido un error y el equipo de Alura ya ha sido notificado...

Si necesita ayuda inmediata, por favor contáctenos. Aprovecha la oportunidad de explorar nuestros cursos :)

[Ver cursos](#)[Voltar para el inicio](#)

## NAVEGACIÓN

PLANES  
INSTRUCTORES  
BLOG  
POLÍTICA DE PRIVACIDAD  
TÉRMINOS DE USO  
SOBRE NOSOTROS  
PREGUNTAS FRECUENTES

## ¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

## BLOG

PROGRAMACIÓN  
YOUTUBE FACEBOOK LINKEDIN  
FRONT END  
DATA SCIENCE  
INNOVACIÓN Y GESTIÓN  
DEVOPS

AOVS Sistemas de Informática S.A  
CNPJ 05.555.382/0001-33

## SÍGUENOS EN NUESTRAS REDES SOCIALES



## ALIADOS

# Empresa participante do SCAL & P ENDEAVOR

En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

**POWERED BY**

## CURSOS

### Cursos de Programación

Lógica de Programación | Java

### Cursos de Front End

HTML y CSS | JavaScript | React

### Cursos de Data Science

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

### Cursos de DevOps

Docker | Linux

### Cursos de Innovación y Gestión



